UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/873,875	06/04/2001	Christophe de Dinechin	10011596-1	5117

22879 7590 11/16/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
VO, LILIAN	
ART UNIT	**PAPER NUMBER**
2195	
MAIL DATE	**DELIVERY MODE**
11/16/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/873,875	DE DINECHIN ET AL.	
	Examiner	Art Unit	
	Lilian Vo	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 August 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 - 26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 - 26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 26 are pending.

2. In view of the appeal brief filed on 8/24/07, PROSECUTION IS HEREBY REOPENED. New grounds of rejection are set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) request reinstatement of the appeal.

If reinstatement of the appeal is requested, such request must be accompanied by a supplemental appeal brief, but no new amendments, affidavits (37 CFR 1.130, 1.131 or 1.132) or other evidence are permitted. See 37 CFR 1.193(b)(2).

Claim Objections

3. **Claim 26** is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. Claim 26 recites the context is stored in memory other than the inconsequential register, which depends on claim 1 claiming to save the context using an inconsequential register. The uses of the memory

other than the inconsequential register as recited in claim 26 is contradicting with its parent claim. See also MPEP. 608.01 (n), "Infringement Test" for dependent claims. The test for a proper dependent claim is whether the dependent claim includes every limitation of the parent claim. The test is not whether the claims differ in scope. A proper dependent claim shall not conceivably be infringed by anything which would not also infringe the basic claim.

Perhaps, applicant intends to claim the context is stored in the memory and the inconsequential register is used to store the address of where the context is being stored. Then, applicant should amend the claim language to clarify the issues.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claim 22 is rejected under 35 U.S.C. 101 because they are directed to non-statutory subject matter.

The language of independent claim 22 raises a question as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practical application producing a useful, concrete and tangible result to form the basis of statutory subject matter under 35 U.S.C. 101.

Independent claim 22 does not appear to require any computer hardware to implement the claimed invention. The claim appears to define the metes and bounds of an invention comprised of software alone. There is no support (i.e., explicitly claimed computer hardware) in

the body of the claims. The "system" of claim 22 appears to be a system comprised entirely of software. Software alone, without a machine, is incapable of transforming any physical subject matter by chemical, electrical, or mechanical acts. If the "acts" of a claimed process manipulate only numbers, abstract concepts or ideas, or signals representing any of the foregoing, the acts are not being applied to appropriate subject matter. In re Schrader, 22 F.3d 290 at 294-95, 30 USPQ2d 1455 at 1458-59 (Fed. Cir. 1994). Transformation of data by a machine constitutes statutory subject matter if the claimed invention as a whole accomplishes a practical application. That is, it must produce a "useful, concrete and tangible result." State Street, 149 F.3d 1368, 1373, 47 USPQ2d 1596 at 1600-02 (Fed. Cir. 1998). MPEP 2106. State Street required transformation of data by a machine before it applied the "useful, concrete, and tangible test." However, State Street does not hold that a "useful, concrete and tangible result" alone, without a machine, is sufficient for statutory subject matter. State Street, 149 F.3d at 1373, 47 USPQ2d at 1601.

Claim 22 is rejected under 35 U.S.C. 101 because the claimed invention, appearing to be comprised of software alone without claiming associated computer hardware required for execution, is not supported by either a specific and substantial asserted utility (i.e., transformation of data) or a well established utility (i.e., a practical application).

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claim 25 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 25 recites the inconsequential register does not store context at a PIP, which depending on claim 1 claiming to store context using the inconsequential register. Claim language as recited is contradicting with its parent claim. According to application specification the inconsequential register is not being used by the host but it is being used to store the context. Appropriate clarification is required.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1, 2, 3, 9 and 22 - 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chatterjee et al. (US 5,634,046, hereinafter Chatterjee).

10. Regarding **claim 1,** Chatterjee discloses a method of switching context on a processor (abstract, col. 2 lines 39 - 52), the method comprising:

saving the context under software control using an inconsequential register (col. 2 lines 35 – 67, col. 8 lines 60 - 67); and

preventing the processor from changing the context while the context is being saved (col. 3 lines 9 - 17).

With respect to the term "inconsequential register", the specification disclosed that it was used as a temporary storage. Since the specification did not provide a precise meaning of an inconsequential register beside it's a type of storage and that it is "a register that is not used at a predetermined interruption point (PIP)", the claim must be given its broadest reasonable interpretation. This means that the words of the claim must be given their plain meaning unless applicant has provided a clear definition in the specification (See MPEP 2111.01.) In this case, the inconsequential register is just a temporary storage. As such, Chatterjee discloses that a stack pointer register is made available for general purpose use by programs for storing operands and results of the operations other than stack pointer itself during context switching without potentially causing stack corruption (col. 7 lines 9 – 29). Also the stack pointer register is utilized to store and retrieve data in the "scratch" space portion of the application program's stack segment (col. 8 lines 60 - 67). Therefore, it would have been obvious for one of an ordinary skill in the art, at the time the invention was made to use the stack pointer register of Chatterjee in place of the claimed inconsequential register during context switching as it can perform the intended functions equally well without causing the system to crash.

11. Regarding **claim 2,** Chatterjee discloses the inconsequential register is used as a temporary storage in lieu of a privileged register (col. 8 lines 60 - 67).

12. Regarding **claim 3**, Chatterjee discloses the context is saved at a predetermined interruption point (abstract, col. 8 lines 27 – 42, col. 2 lines 8 - 21).

13. **Claims 9 and 22** are rejected on the same ground as stated in claim 1 above.

14. Regarding **claim 23,** Chatterjee discloses the content of the inconsequential register is corrupted during the context switch (col. 2 lines 22 – 34, col. 8 lines 60 - 67).

15. Regarding **claim 24,** as modified Chatterjee discloses the inconsequential register includes storing an address, the address indicating a memory location at which the context will be saved (Chatterjee: col. 3 lines 18 - 27, col. 6 line 52 - col. 7 line 30).

16. Regarding **claim 25,** as modified Chatterjee discloses the inconsequential register does not store context a predetermined interruption point (Chatterjee: col. 3 lines 18 - 27, col. 6 line 52 - col. 7 line 30).

17. Regarding **claim 26,** as modified Chatterjee discloses the context is stored in memory other than the inconsequential register (Chatterjee: col. 3 lines 18 - 27, col. 6 line 52 - col. 7 line 30).

18. Claims 4 – 5, 11 - 16 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chatterjee et al. (US 5,634,046) as applied to claim 1 above, in view of Bugion et al. (US 6,496,847, hereinafter Bugion).

19. Regarding **claim 4,** Chatterjee discloses the context switching between the execution entities (abstract, col. 6 line 52 – col. 7 line 8). Bugion discloses the context is switched between a host operating system and a virtual machine application (Bugion: col. 4, lines 52 – 61: switching from HOS context to VMM context. Col 11, lines 30 – 52), the virtual machine application controlling the context switch (col. 11, lines 30 – 52: VMM handles directly all exceptions and interrupts that occur while executing in the VMM context. Col. 12, lines 20 – 24: the VMM completely takes over the machine and only voluntarily relinquishes control to the HOS). Therefore, it would have been obvious for one of an ordinary skill in the art, to relate Bugion's teaching together with Chatterjee to use stack pointer register as a form of temporary storage to perform the context switching between the execution entities as required because the context switch of Bugion can be performed successfully without being crashed as stated in col. 11, lines 30 – 52 (total switch preferably first saves the state before setting it according to the target context in order to facilitate the inverse operation) and col 17, lines 18 – 21 (ensure that no interrupts occur during the switch).

20. Regarding **claim 5,** as modified Chatterjee discloses the use of inconsequential register is used to pass information to the virtual machine application (Chatterjee: col. 2 lines 35 – 67, col. 8 lines 60 – 67. Bugion: col. 11, lines 30 – 52, col. 16, lines 45 - 61).

21. Regarding **claim 11,** Chatterjee discloses a method of switching context between executing programs on a processor (abstract), the processor having privileged registers (col. 1 lines 5 – 7, 45 – 52: special purpose registers store data relating to control, exceptions, memory management and the like. Col. 1 line 54 – col. 2 line 7), the processor having access to other memory (fig. 1, col. 4 lines 9 - 32), the method comprising:

giving the program access to the privileged registers (Col. 1 line 54 – col. 2 line 7, col. 6 line 51 – col. 7 line 29);

using at least one privilege register as temporary storage to save the context in other the other memory at a predetermined interruption point (col. 6 line 51 – col. 7 line 29); and

preventing the processor from changing the context while the context is being saved (col. 3 lines 9 – 17, col. 7 lines 9 – 29);

the program controlling the context switch (abstract, col. 8 lines 27 – 42, col. 2 lines 8 - 21).

Chatterjee discloses the context switching between the execution entities (abstract, col. 6 line 52 – col. 7 line 8). Bugion discloses the context is switched between a host operating system and a virtual machine application (Bugion: col. 4, lines 52 – 61: switching from HOS context to VMM context. Col 11, lines 30 – 52), the virtual machine application controlling the context switch (col. 11, lines 30 – 52: VMM handles directly all exceptions and interrupts that occur while executing in the VMM context. Col. 12, lines 20 – 24: the VMM completely takes over the machine and only voluntarily relinquishes control to the HOS). Therefore, it would have been obvious for one of an ordinary skill in the art, to relate Bugion's teaching together with

Chatterjee to perform the context switching between the execution entities and to use stack pointer register as a form of temporary storage to store the context because the context switch of Bugion can be performed successfully without being crashed as stated in col. 11, lines 30 – 52 (total switch preferably first saves the state before setting it according to the target context in order to facilitate the inverse operation) and col. 17, lines 18 – 21 (ensure that no interrupts occur during the switch).

22. **Claims 12 – 16 and 20** are rejected on the same grounds as stated in claims 1 – 5, 9 and 11 above.

23. Claims 6 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chatterjee et al. (US 5,634,046), as applied to claim 1 above, and in view of Applicants' admitted prior art (hereinafter AAPA).

24. Regarding **claim 6,** Chatterjee did not clearly disclose the context switched is using an IA-64 processor. However, an IA-64 processor is considered a well-known architecture as disclosed in Applicants' admitted prior art (specification page 1, paragraph 4). It would have been obvious for one of an ordinary skill in the art, to implement Chatterjee system with an IA-64 processor and still able to perform the intended functions equally well without causing the system to crash because Chatterjee discloses that other processors which provide a stack pointer register and protected mode operation also may be suitable for use with his invention (col. 4 lines 5 – 8).

25. Regarding **claim 7,** Chatterjee discloses the inconsequential register is a caller-save register (col. 8 lines 60 – 67).

26. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chatterjee et al. (US 5,634,046), as applied to claim 1, in view of Applicants' admitted prior art, and further in view of Yamaura et al. (US. Pat. Application Publication 2001/0008563, hereinafter Yamaura).

27. Regarding **claim 8,** as modified Chatterjee did not clearly disclose the inconsequential registers is a branch register. Nevertheless, Yamaura discloses a system that use link register (caller-save register) for holding an address of a source from which a subroutine call is made and LI and LN registers (branch register, caller-save register) for holding branch destination addresses at a time of IRQ (page 6, paragraph 0110 and page 12, paragraph 209, page 5, paragraph 0082). It would have been obvious for one of an ordinary skill in the art, at the time the invention was made, to particularly implement as modified Chatterjee's system with Yamaura's teaching with the use of link register and/or branch register as a storage to hold the addresses of the current context information so that data can be accessed more quickly.

28. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chatterjee et al. (US 5,634,046), as applied to claim 1 above, and in view of Yamaura et al. (US. Pat. Application Publication 2001/0008563).

29. Regarding **claim 10,** Chatterjee did not clearly disclose the context is restored by using a branch register to perform an indirect branch. Nevertheless, Yamaura disclose a system that use a plurality of registers for storing data to undergo operation processing which can be freely written/read to/from the registers (page 5, paragraph 0082). Furthermore, Yamaura discloses that LI and LN registers (branch register, caller-save register) are used for holding branch destination addresses at a time of IRQ (page 6, paragraph 0110 and page 12, paragraph 209, page 5, paragraph 0082). It would have been obvious for one of an ordinary skill in the art, at the time the invention was made, to particularly implement Chatterjee's system with Yamaura's teaching with the use of branch register as a storage to hold the destination addresses at a time of IRQ when restoring the context so that processing can be resumed from last interrupt efficiently and quickly.

30. Claims 17 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chatterjee et al. (US 5,634,046) in view of Bugion et al. (US 6,496,847), as applied to claim 12 above, and further in view of Applicants' admitted prior art.

31. Regarding **claim 17,** as modified Chatterjee did not clearly disclose the context switched is using an IA-64 processor. However, an IA-64 processor is considered a well-known architecture as disclosed in Applicants' admitted prior art (specification page 1, paragraph 4). It would have been obvious for one of an ordinary skill in the art, to implement as modified Bugion's system with an IA-64 processor and still able to perform the intended functions equally well without causing the system to crash because Chatterjee discloses that other processors

which provide a stack pointer register and protected mode operation also may be suitable for use with his invention (col. 4 lines 5 – 8).

32. Regarding **claim 18,** Chatterjee discloses the inconsequential register is a caller-save register (col. 8 lines 60 – 67).

33. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chatterjee et al. (US 5,634,046) in view of Bugion et al. (US 6,496,847), as applied to claim 12 above, in view of Applicants' admitted prior art and further in view of Yamaura et al. (US. Pat. Application Publication 2001/0008563).

34. Regarding **claim 19,** as modified Chatterjee did not clearly disclose the temporary storage includes caller-save register or branch register. Nevertheless, Yamaura discloses a system that use link register (caller-save register) for holding an address of a source from which a subroutine call is made and LI and LN registers (branch register, caller-save register) for holding branch destination addresses at a time of IRQ (page 6, paragraph 0110 and page 12, paragraph 209, page 5, paragraph 0082). It would have been obvious for one of an ordinary skill in the art, at the time the invention was made, to particularly implement as modified Chatterjee's system with Yamaura's teaching with the use of link register and/or branch register as a storage to hold the addresses of the current context information so that data can be accessed more quickly.

35. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chatterjee et al. (US 5,634,046) in view of Bugion et al. (US 6,496,847), as applied to claim 12 above, and further in view of Yamaura et al. (US. Pat. Application Publication 2001/0008563).

36. Regarding **claim 21,** as modified Chatterjee did not clearly disclose the context is restored by using a branch register to perform an indirect branch. Nevertheless, Yamaura disclose a system that use a plurality of registers for storing data to undergo operation processing which can be freely written/read to/from the registers (page 5, paragraph 0082). Furthermore, Yamaura discloses that LI and LN registers (branch register, caller-save register) are used for holding branch destination addresses at a time of IRQ (page 6, paragraph 0110 and page 12, paragraph 209, page 5, paragraph 0082). It would have been obvious for one of an ordinary skill in the art, at the time the invention was made, to particularly implement as modified Chatterjee's system with Yamaura's teaching with the use of branch register as a storage to hold the destination addresses at a time of IRQ when restoring the context so that processing can be resumed from last interrupt efficiently and quickly.

Response to Arguments

37. Applicant's arguments with respect to claims 1, 11, 12 and 22 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lilian Vo whose telephone number is 571-272-3774. The examiner can normally be reached on Thursday 8am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Lilian Vo
Examiner
Art Unit 2195

ln
November 8, 2007

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100